PointRas: Uncertainty-Aware Multi-Resolution Learning for Point Cloud Segmentation

Yu Zheng, Xiuwei Xu, Jie Zhou, *Senior Member, IEEE*, and Jiwen Lu, *Senior Member, IEEE*

***Abstract*— In this paper, we propose an uncertainty-aware multi-resolution learning for point cloud segmentation, named PointRas. Most existing works for point cloud segmentation design encoder networks to obtain better representation of local space in point cloud. However, few of them investigate the utilization of features in the lower resolutions produced by encoders and consider the contextual learning between various resolutions in decoder network. To address this, we propose to utilize the descriptive characteristic of point clouds in the lower resolutions. Taking reference to core steps of rasterization in 2D graphics where the properties of pixels in high density are interpolated from a few primitive shapes in rasterization rendering, we use the similar strategy where prediction maps in lower resolution are iteratively regressed and upsampled into higher resolutions. Moreover, to remedy the potential information deficiency of lower-resolution point cloud, we refine the predictions in each resolution under the criterion of uncertainty selection, which notably enhances the representation ability of the point cloud in lower resolutions. Our proposed PointRas module can be incorporated into the backbones of various point cloud segmentation frameworks, and brings only marginal computational cost. We evaluate the proposed method on challenging datasets including ScanNet, S3DIS, NPM3D, STPLS3D and ScanObjectNN, and consistently improve the performance in comparison with the state-of-the-art methods.**

***Index Terms*— Point cloud segmentation, semantic segmentation, multi-resolution learning, contextual learning.**

I. Introduction

RECENT years have witnessed the rocketing development of 3D sensors, such as the lidar scanners and the light-field cameras. The easy access to the 3D point clouds enables the machines to better perceive the real-world environment and benefit 3D-based applications such as virtual reality, autonomous driving and 3D gaming. The task of 3D point cloud segmentation, which aims at recognizing the labels of individual 3D points, is crucial to the functionality of those applications. However, due to the ununiform density and lack of structure of the data modality, directly utilizing the general convolutional methods is non-trivial.

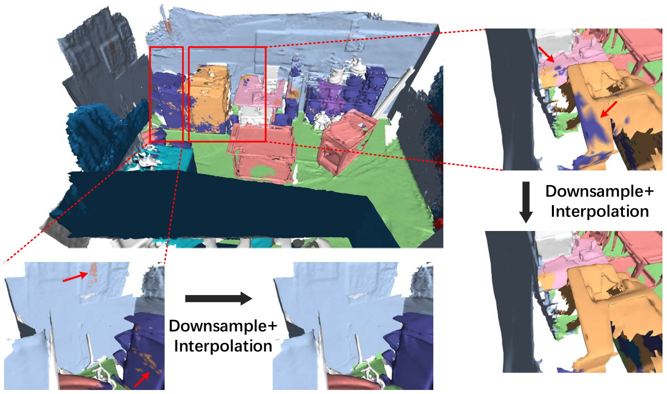

Fig. 1. An example scene predicted by PointConv [1] from the ScanNet [2] v1 test dataset, where 2 independent pieces are displayed in detail. For higher-resolution predictions with labeling errors (red arrows), we downsample the predictions using furthest point sampling ($\times\frac{1}{128}$), and upsample them into the raw resolution via nearest neighbor interpolation. In the illustrated example, we keep off the noise-like errors and preserve the semantic information to a large extent through downsampling and interpolation.

Compared to the discretization-based [3] or projection-based [4] methods which require high computational cost or cause information loss in geometry, the pioneer work PointNet [5] firstly consumes the raw point clouds. PointNet++ [6] constructs hierarchical 3D receptive field and extracts the multi-resolution features using the set abstraction (SA) layer. Based on this hierarchical architecture, the following works attempt to enhance the point-wise feature [1], [7], [8], structuralize the points in local regions using kernel functions [9], [10] or explore the non-spherical local shapes [11], [12]. Compared to the 2D feature maps which lack visual interpretability in deep conv-layers [13], the downsampled point coordinates across various resolutions partially preserve the structure of 3D shapes. Moreover, imperfections are prone to be tolerated under lower-resolution conditions, as shown in Fig. 1. It is due to the rich texture and geometry features condensed in the data modality of 3D point clouds. For example, the depth information is well preserved after subsampling in a RealSense camera [14]. However, most existing works for point cloud segmentation aim to design encoder networks and obtain better representation of local space in point cloud, or apply post-processing methods upon the raw highest resolution as regularization [15], [16]. Few of them investigate

Manuscript received 21 July 2021; revised 24 June 2022 and 29 August 2022; accepted 29 August 2022. Date of publication 14 September 2022; date of current version 19 September 2022. This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFA0700802, in part by the National Natural Science Foundation of China under Grant 62125603 and Grant U1813218, and in part by a Grant from the Beijing Academy of Artificial Intelligence (BAAI). The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Joao M. Ascenso. *(Corresponding author: Jiwen Lu.)*

The authors are with the Beijing National Research Center for Information Science and Technology (BNRist), Department of Automation, Tsinghua University, Beijing 100084, China (e-mail: zhengyu19@mails.tsinghua.edu.cn; xxw17@mails.tsinghua.edu.cn; jzhou@tsinghua.edu.cn; lujiwen@tsinghua.edu.cn).

Digital Object Identifier 10.1109/TIP.2022.3205208

the utilization of features in lower resolution produced by the encoder and consider the contextual learning between various resolutions in decoder network. Moreover, recent work claims that the extracted information in lower-resolution points is overwhelmed by information in higher-resolution under specific measurements [17]. The utilization of the lower-resolution point clouds is limited.

In this paper, we propose a efficient but effective point rasterization (PointRas) method, which exploits the representation ability of lower-resolution point clouds. While not strictly following the procedures of graphical point-cloud rasterization, PointRas takes reference to two core steps in rasterization: (1) determine the pixels covered by a primitive shape; (2) interpolation of the output attributes (such as color and light) for all covered pixels, as shown in Fig. 2(b). The primitives provide the geometric sketch for the rasterization step, where the geometric skeleton is mapped into the continuous fragments using triangle interpolation. Then the attributes of the dense grids such as color and light are determined through further processing, which is analogous to the final point-wise labeling in semantic segmentation. In our case, we can utilize the lower-resolution representation of point clouds to depict the rough structures of a whole scan. For those object entities with plain geometric features (e.g., the surface of a wall plane), the lower-resolution descriptors can well depict their properties, while the spatial redundancy exists in higher-resolution representations [18] which may produce noise-like errors (as shown in Fig. 1). Therefore, we directly regress the point-wise segmentation scores in a low resolution. To further utilize their representation ability, prediction scores in lower resolution are then interpolated into higher resolutions. Moreover, to tackle the potential information deficiency merely in lower-resolution point cloud, we adopt the strategy of uncertainty selection inspired by 2D cases [19], where a certain proportion of points are re-predicted as refinement in each resolution. The results after all the iterations finally refine the predictions by the decoder of baseline network in the highest resolution. The proposed PointRas module is computationally efficient and brings only marginal FLOPs compared to the original baseline network. We also evaluate the proposed method on the challenging ScanNet, S3DIS, NMPM3D, STPLS3D and ScanObjectNN datasets, and consistently improve the performance in comparison with the state-of-the-art methods.

We summarize key contributions of our work as follows:

1) We propose a point rasterization (PointRas) method which iteratively labels and upsamples the lower-resolution point clouds whilst preserving the structures of the raw backbone network, thereby utilizing and enhancing the representation ability of point clouds across various resolutions.

2) To remedy the potential information deficiency of lower-resolution point clouds, we propose to refine the predictions in each resolution under the criterion of uncertainty selection, whose location do not depend on extra supervision signals.

3) We conduct extensive experiments on challenging datasets including ScanNet [2], S3DIS [20], NPM3D [21], STPLS3D [22] and ScanObjectNN [23] to demonstrate the efficacy of PointRas. The results show that the proposed method is notably effective and adaptive to various state-of-the-art methods, bringing only marginal computational costs.

II. Related Work

In this section, we discuss related work in three aspects: point cloud segmentation, decoder architectures for point cloud recognition, and contextual learning in multiple resolutions.

A. Point Cloud Segmentation

Current approach to point cloud segmentation can be categorized into 3 divisions: projection-based, discretization-based and point-based. The **projection-based** approaches utilize the efficient 2D convolution method after projecting point clouds into single [24], [25] or multiple [4], [26] 2D planes, which inevitably results in losses of 3D spatial information. The **discretization-based** methods voxelize unstructured point cloud into regular 3D grids and apply 3D convolutions [3], [27]. The 3D UNet architectures are adopted as backbones for downstream tasks such as semantic segmentation [3] and instance segmentation [28], [29], [30]. However, the computational cost and performance of discretization-based methods is sensitive to spatial granularity. Spatial information is also lost during quantization of 3D coordinates. As pioneering work of **point-based** methods, PointNet [5] consumes raw points using multi-layer perceptron (MLP) in semantic segmentation. PointNet++ [6] then attempts to exploit the local features via subsampling and kNN grouping. The following works further enhance the point-wise features [1], [7], [8], [31], structuralize the points in the local regions using kernel functions [9], [10], [32], explore the non-spherical local shapes [11], [12] and attention mechanism [33] in point cloud. Other methods include the graph-based message passing [34], [35], the embedding of the point clouds into an implicit distance field [36], projection onto 2D planes [37], [38]. Recent works overcome the inefficiency of the neighbor searching mechanism, such as random query of the neighboring points [39], [40], [41]. We favor the point-based methods where spatial information is largely preserved by efficient point-wise multi-layer perceptron (MLP) or kernal function. Moreover, the explicit downsampling (e.g., furthest point sampling [6], grid sampling [10] and random sampling [39]) operation in point-based methods provides us with hierarchical representations of point cloud in various resolutions. For more in-depth acquaintance with relevant researches, we strongly recommend referring to the literature review in [42].

B. Decoder Architectures for Point Cloud Recognition

The encoder-decoder structures are widely adopted for point cloud segmentation [1], [6], [10], [34] and generation [43], [44], [45]. Given the latent, high-dimensional and low-resolution features of point cloud via the encoder network with point downsampling (PD), the decoder network aims to progressively recover the representations in high resolutions by feature upsampling (FU), especially for point cloud

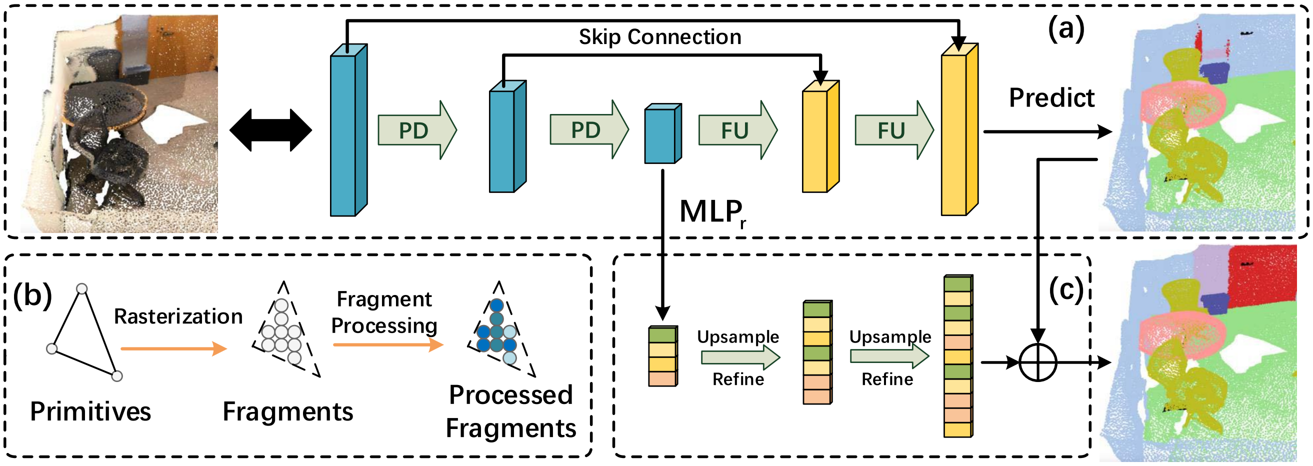

Fig. 2. The framework of PointRas for segmentation refinement. **(a)** The point downsampling (PD) and feature upsampling (FU) pipeline of the backbone network for 3D point cloud segmentation, such as PointNet++, PointConv and KPConv. **(b)** Parts of the procedures for 3D rendering as a comparison with our method. The rasterization step maps the features of the primitives to the pixels or coordinates in higher density, called fragments. Then the attributes are determined after further processing. **(c)** Our proposed **PointRas Module**, which iteratively interpolates higher-resolution "fragments" and refines them. After the last iteration, the refined prediction scores in PointRas partially replace (⊕ in **(c)**) the ones at the same positions predicted by the backbone, as shown in the final predictions at the bottom-right corner.

segmentation task which aims to predict per-point labels. PointNet++ [6] pioneerly proposes the decoder based on feature propagation layer, which hierarchically interpolates the bottleneck features to the higher resolution weighted by the inverse distance from centroid to neighboring points. The point-wise labels are not predicted until the raw resolution is recovered. The following works mainly focus on the convolution kernels in encoder to obtain better bottleneck features, while the basic idea of feature upsampling is largely preserved in their decoder architectures [1], [8], [10], [34]. In point cloud completion, a classic folding-based decoder network proposed in [43] deforms a canonical 2D square onto the implicit 3D object surface of a point cloud, which is able to reconstruct arbitrary point cloud shapes. The folding-based pattern is preserved in other following variants such as AtlasNet [46] and DeepSDF [47]. In this work, we focus on the task of point cloud segmentation in more complex environments, and utilize the features in lower resolutions produced by decoder network to exploit better representation ability.

C. Contextual Learning in Multiple Resolutions

Previous works on contextual learning in point cloud segmentation usually focus on the relation between points [7], [40], [48], [49] or interaction between local regions [8] under fixed spatial granularity. However, the downsampling operation [6], [39], [50] naturally provides hierarchical representation of point cloud in various resolutions. Following 2D cases [51], [52], the skip connection between different resolutions is used in 3D segmentation networks [53], [54]. The architecture of feature pyramid network [45], [55] combines the higher-resolution, semantically weak features with lower-resolution, semantically strong features using lateral connection. DRNet [56] proposes an error-minimizing module to enhance the local feature representation in different resolutions, which is tested on the datasets with individual and simple objects. For sequential data, the lateral connection is used for fusing the information between different temporal resolutions [57]. The contextual neighboring information shows its effectiveness naturally in point cloud upsampling [58]. The graph structure models the dependency between the multi-resolution supervision signals [59]. Instead of using connection as feature enhancement, PointRend [19] interpolates the lower-resolution representations under the guidance of the higher-resolution grids. [19] utilizes the mechanism that in rendering, the properties of 2D pixels are mapped from the continuous entity. Our method intends to enhance the representation ability in different resolutions for 3D scenes. In our case, the predicted segmentation scores in higher resolution are interpolated from neighbors in the lower resolution, which resembles the rasterization operation in graphical rendering.

III. Approach

The general pipeline of point cloud segmentation contains a backbone network for both feature encoding and decoding. The backbone downsamples the features of point clouds and propagates them to the higher-resolution points. As shown in Fig. 2, we propose a point rasterization (PointRas) module to iteratively refine the segmentation scores in various resolutions. The module is suitable for the general backbone networks with such sampling mechanism for point cloud segmentation.

A. Preliminaries

In the basic pipeline of 3D rendering, the transformed vertices are assembled into primitives (as shown in Fig. 2(b), 2 vertices into lines, 3 vertices into triangles, etc.). Then the rasterization operation interpolates the values across the primitives into pixel fragments [60], enumerating the pixels encapsulated by those primitives. The attributes of fragments, such as the light and color are determined after being shaded. For computer graphics, the information contained in the vertices is crucial for figuring out what objects look like in the digital world, because the shapes are generally displayed as a collection of lines, planes, etc.

The general backbone for point cloud segmentation contains a hierarchical architecture, which encodes the rich geometric features in the downsampled points and propagates them into higher resolution. We can regard the downsampled points as the "vertices" of shapes in raw point clouds, and semantic or instance label of the individual points as the results of fragment shading. Due to the rich texture and geometry features condensed in 3D point clouds, the downsampled point clouds across various resolutions partially preserve the structure of 3D shapes. Meanwhile, the lower-resolution data tolerates more imperfection such as the spatial noises. To this end, we attempt to better exploit the representation ability of those lower-resolution points. PointRas interpolates the features of lower-resolution points to the higher ones, utilizing and enhancing their representation ability with limited information. We illustrate the baseline method incorporated with our proposed module in Fig. 2(a) and Fig. 2(c).

B. PointRas Module

Suppose in one decoder stage, the number of lower-resolution and higher-resolution points are N_{lr} and N_{hr}, respectively. As initialization, we regress the prediction scores M_{lr} from features F_{lr} in a low-resolution layer of the backbone decoder using a learnable MLP (MLP_r in Fig. 2):

$$M_{lr} = MLP_r(F_{lr}). \tag{1}$$

Then for each decoder stage in the backbone network, i.e., in each iteration of PointRas that follows, we conduct 2 key steps named **feature upsampling** and **prediction updates**.

1) Feature Upsampling: As the original backbone structuralizes the raw point clouds in advance using sub-sampling methods such as block sampling [6] and grid sampling [10], we could already obtain the 3D coordinates of both the lower-resolution and higher-resolution points before applying PointRas. Given features F_{lr} and segmentation predictions M_{lr} in lower resolution, we upsample them into a higher resolution via interpolation based on the correspondence between different resolutions. Specifically, from the coordinates of lower-resolution points, we query k nearest neighbor points $x_{lr}^{(1)}, x_{lr}^{(2)}, \ldots, x_{lr}^{(k)}$ for each higher-resolution point $x_{hr}^{(j)}$ as $N(x_{hr}^{(j)})$, forming the "vertices" of PointRas. The situations of $k = 1$ and $k>1$ correspond to the nearest neighbor interpolation used in [10] and [39] and the inverse distance weighted interpolation used in [1], [6] respectively. For $k > 1$, as shown in (2), the higher-resolution prediction scores $\hat{M}_{hr}$ and features F_{hr} are interpolated using the corresponding "vertices" in lower resolution (subscripted with lr). The weight of interpolation is negatively correlated to the distance between 2 points, denoted as $d(\cdot)$, which means the closer "vertices" in lower resolution contribute more to the feature representation of higher-resolution points than those far-away "vertices".

$$F_{hr}^{(j)} = \frac{\sum_{i=1}^{k} w_{ij} F_{lr}^{(i)}}{\sum_{i=1}^{k} w_{ij}}, \quad \hat{M}_{hr}^{(j)} = \frac{\sum_{i=1}^{k} w_{ij} M_{lr}^{(i)}}{\sum_{i=1}^{k} w_{ij}}$$

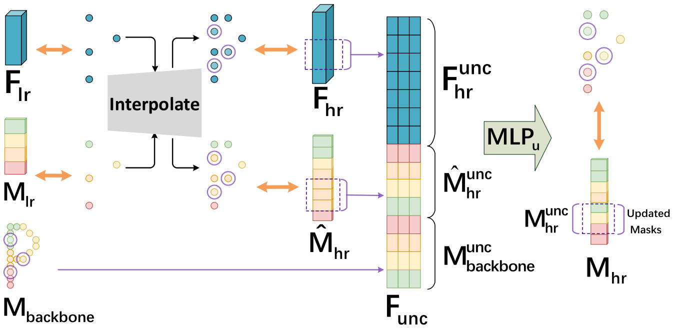

Fig. 3. One PointRas iteration. The lower-resolution segmentation predictions M_{lr} and features F_{lr} are interpolated into higher-resolution ($\hat{M}_{hr}$ and F_{hr}). N_{unc} points are selected from $\hat{M}_{hr}$ as uncertain predictions. The interpolated features and prediction scores, along with the scores predicted by the baseline backbone of those N_{unc} points are concatenated and re-predicted by MLP_u. The bidirectional arrow indicates the equivalence relation. We set the number of selected points as 3 for easy illustration, which is indicated by the horizontal dimension of F_{unc}.

$$\text{where} \quad w_{ij} = \frac{1}{d(x_{lr}^{(i)}, x_{hr}^{(j)})^2}, \; x_{lr}^{(i)} \in N(x_{hr}^{(j)}) \tag{2}$$

Note that for each resolution, the encoder network in the backbone has already queried the indices of k nearest neighbors for $x_{hr}^{(j)}$. Therefore, the upsampling in PointRas saves the operation of k-NN algorithm with the complexity of $O(N_{lr}N_{hr})$. By default, we use the same interpolation method as the backbone decoder. We show in the Experiment section that the interpolation method applied in PointRas can differ from backbone decoder, which demonstrates the flexibility of the proposed method.

2) Prediction Updates: The features in lower-resolution provide the informative "primitives" for point cloud in higher resolution. However, as claimed in [61], some fine-grained geometric features are lost during the rasterization process. Therefore, potential uncertainty caused by information deficiency exists in the predictions of $\hat{M}_{hr}$, as the information is limited simply by interpolating the values in the initial lower resolution. It is necessary to update the prediction scores in PointRas as refinement. To locate those point scores predicted with uncertainty, we analyze the predicted scores through the score-based self digging. For each point score in $\hat{M}_{hr}$, we calculate the **score difference** between its most probable category and the runner-up category. We select N_{unc} points from $\hat{M}_{hr}$ with the least **score difference** as the uncertain predictions. Those selected points are illustrated as the circled points in Fig. 3. We concatenate their interpolated scores and features, along with the prediction scores gathered from the original prediction by the backbone decoder ($M_{backbone}$ in Fig. 3), each denoted as F_{hr}^{unc}, $\hat{M}_{hr}^{unc}$ and $M_{backbone}^{unc}$. The concatenated feature are fed into another multi-layer perceptron MLP_u to renew the uncertain predictions. The N_{unc} updated prediction scores M_{hr}^{unc} are inserted back to $\hat{M}_{hr}$ and replace the previous prediction scores at corresponding indices, generating the updated higher-resolution predictions M_{hr}. The process of prediction updating is summarized as follows:

$$\begin{aligned} M_{hr}^{unc} &= MLP_u([F_{hr}^{unc}, \hat{M}_{hr}^{unc}, M_{backbone}^{unc}]) \\ M_{hr} &= [M_{hr}^{unc}, \hat{M}_{hr}^{-unc}] \end{aligned} \tag{3}$$

where $\hat{M}_{hr}^{-unc}$ denotes the exclusion of uncertain predictions $\hat{M}_{hr}^{unc}$ from $\hat{M}_{hr}$. $[\cdot]$ denotes the operation of element-wise concatenation along the feature dimension.

Apart from the aforementioned self digging approach, the uncertain predictions can be located in a **comparing** manner. For each point in $\hat{M}_{hr}$, we can calculate its norm-based **score difference** with the corresponding point in $M_{backbone}$ and similarly, select N_{unc} points with the least **score differences**. By default, we adopt the **self digging** method and we will discuss the **comparing** method at the experiment section.

3) Iterative Rasterization: The proposed PointRas module iteratively applies the iterative operations illustrated in Fig. 3. At the end of the current iteration, the interpolated F_{hr} and the renewed M_{hr} are fed into the next iteration, regarded as the new input lower-resolution features and prediction scores (F_{lr} and M_{lr}). To utilize the results of k-NN algorithm provided by the backbone network, each PointRas iteration pairs with a feature upsampling (FU) layer of the backbone. The values of N_{lr} and N_{hr} in one PointRas iteration are kept the same with the number of points before and after the corresponding FU layer. The learnable parameters of MLP_u in Fig. 3 are shared across all iterations. After the last iteration, the resolution of M_{hr} are the same with the raw point clouds. To utilize the prediction scores of PointRas after all iterations, we select the uncertain predictions of $M_{backbone}$ using the same criterion in PointRas iteration. Finally, we replace them with the refined prediction scores of M_{hr} at the same positions. The final replacement is denoted as $\oplus$ in Fig. 2(c).

The implementation of the proposed PointRas module is iterative but computationally efficient and compact. MLP_r is only used for regressing the whole prediction scores for the initial resolution, i.e., before the first PointRas iteration. During inference, such score regression of all points by MLP_r does not exist in the following PointRas iterations. Moreover, the parameters of MLP_u layer are shared across all PointRas iterations when refining the uncertain predictions. In summary, the additional learnable parameters apart from the backbone network only contain one MLP (MLP_r) before the first PointRas iteration, and another shared MLP (MLP_u) across all PointRas iterations for refinement of prediction scores.

C. Implementation Details

In each iteration of PointRas, the proportion of selected uncertain predictions in the higher-resolution ($\frac{N_{unc}}{N_{hr}}$) is fixed as α. After all the PointRas iterations, we select from $M_{backbone}$ a smaller proportion (β) of points as uncertain predictions via self digging, and replace them with the corresponding point scores in M_{hr}. The design of a smaller β aims at maintaining the updating quality of final prediction scores. α and β are set as $\frac{1}{2}$ and $\frac{1}{6}$ respectively as the default setting. Compared to the PointRend [19] for 2D image segmentation, we only use the strategy of uncertainty-based selection and do not randomly sample the extra points as supplement. This is because the downsampling strategy for point clouds has already taken the spatial uniformity into consideration, such as the farthest point sampling (blocking subsampling) and grid subsampling.

Algorithm 1 PointRas: Inference

```
input : Number of iterations $T$, features of initial resolution
        $F_{lr}$, scores predicted by the backbone
        $M_{backbone}$,$MLP_r$, $MLP_u$, $\alpha$, $\beta$.
output: Final updated $M_{backbone}$.
Regress lower-resolution prediction scores:
  $M_{lr} = MLP_r(F_{lr})$.
for $t \leftarrow 1, 2 \dots T$ do
    Get $\hat{M}_{hr}$ and $F_{hr}$ using the interpolation in (2) or
      nearest neighbor interpolation.
    Select the predictions with uncertainty from $\hat{M}_{hr}$ as
      $\hat{M}_{hr}^{unc}$, with the proportion of $\alpha$. Extract the
      corresponding points from $F_{hr}$ and $M_{backbone}$ as $F_{hr}^{unc}$
      and $M_{backbone}^{unc}$.
    Get $M_{hr}$ using the re-prediction in (3).
    $M_{lr} \leftarrow M_{hr}, F_{lr} \leftarrow F_{hr}$.
Select the predictions with uncertainty from $M_{backbone}$, with
  the proportion of $\beta$. Replace them with the corresponding
  points in $M_{hr}$.
```

Algorithm 2 PointRas: Training

```
input : Features of initial resolution $F_{lr}$, scores predicted by
        the backbone $M_{backbone}$.
output: $MLP_r, MLP_u$.
Initialize $MLP_r$ and $MLP_u$.
Regress the prediction scores in the initial resolution:
  $\hat{M}_{lr} = MLP_r(F_{lr})$.
Re-predict all points in $\hat{M}_{lr}$ using (4).
Update $MLP_r$ and $MLP_u$ by minimizing (5).
```

During training, $\hat{M}_{lr}$ in initial resolution are still regressed using MLP_r. However, we directly re-predict all N_{lr} points using MLP_u (as shown in (4)) in initial resolution, which aims to enable the awareness of all point scores for selective refinement [19]. Neither interpolation into higher resolution, nor selection of uncertain predictions, nor replacement for $M_{backbone}$ is implemented in training since all points are covered by MLP_u in initial resolution. During training, the loss in (5) is minimized, which contains the supervision on the prediction scores before and after the re-prediction.

$$M_{lr} = MLP_u([F_{lr}, \hat{M}_{lr}, M_{backbone}(lr)]) \tag{4}$$

$$L_{PointRas} = CE(\hat{M}_{lr}, Y_{lr}) + CE(M_{lr}, Y_{lr}) \tag{5}$$

$[\cdot]$ denotes the operation of element-wise concatenation along the feature dimension. $M_{backbone}(lr)$ denotes the extracted scores from $M_{backbone}$ from the corresponding indices of $\hat{M}_{lr}$. CE is the cross entropy loss function.

For inference, from the initial resolution, each iteration corresponds to a feature upsampling operation in the decoder of backbone network (as shown in Fig. 2). The procedures of inference and training for PointRas are summarized in Algorithm 1 and Algorithm 2 respectively.

Note that different initial resolutions in PointRas might differ a lot in terms of information condensed in M_{lr} and thereby affecting the segmentation performance. For various backbone networks, we uniformly set the initial resolution of PointRas the same with the second decoder layer. In other words, the input to PointRas is in the second-least resolution. We show in the experiment section that compared to this default configuration, the information deficiency in lower resolution or redundancy in higher resolution might harm the performance.

TABLE I provides a summary of architectures and configurations used for the backbone networks in this paper. The set abstraction (SA) layers query the neighbor points within a fixed radius, i.e., single scale grouping (SSG). The MLP layers contain batch normalization and learnable biases which are not shown in the table. We also implement BA-GEM [62] equipped with PointRas which incorporates boundary information to encode point-wise features, whose encoder-decoder architecture is upgraded from PointConv [1] with additional supervision signals. These architectures and configurations were fixed before and after being incorporated with PointRas.

D. Discussion

1) PointRend v.s. PointRas: Both PointRend and PointRas underline the usage of multi-resolution information. The difference between PointRend and PointRas can be listed below:

- The strategy of selecting uncertain prediction is supervised in PointRend while our PointRas is unsupervised. Specifically, PointRend uses the distance (less distance indicates more uncertainty) between 0.5 and the probability of the ground truth class in the prediction as the point-wise uncertainty measure. PointRas uses the score difference (less difference indicates more uncertainty) between the largest and the runner-up, which does not depend on the ground truth label. Considering the heavy labeling labor of 3D point clouds, the proposed metric is a promising strategy in the future.
- For the segmentation task (especially for instance segmentation), PointRend is employed upon local regions defined by the object proposals. However, as the point cloud data is unstructured, obtaining such local regions in 3D space is less efficient than gridded images. Instead, the usage of PointRas is not constrained by such local regions. For example, in 3D-BoNet [63], PointRas is implemented in the whole feature map at the backbone network rather than the proposal layers.
- We adopt the inverse distance weighted interpolation in PointRas, which considers the importance of neighboring information by weighing the distance in continuous Euclidean space, therefore being suited for point cloud data. Instead, the bilinear interpolation used in PointRend and trilinear interpolation are more suitable for grid space.

2) Feature Propagation Layer v.s. PointRas: Similar to the feature propagation layer proposed in [6], the inverse distance weighted interpolation in (2) can be adopted to obtain the higher-resolution representations. However, the difference is that we introduce the prediction and refinement of point-wise scores in each lower resolutions, while the feature propagation layer does not further utilize these bottleneck features. Besides, PointRas does not contain any skip connection from the point downsampling layers in the encoder network. Instead, PointRas exploits the representation ability of various resolutions from initial limited information. Interpolation-based upsampling and analysis on prediction scores are adopted to fully exploit its semantic information. Moreover, PointRas can be adapted to different interpolation methods, which will be discussed in the experiment section.

3) Rasterization in Rendering v.s. PointRas: Our PointRas does not strictly follows the procedures of point-cloud rasterization, as our task aims at per-point labeling in 3D space which does not require the projection onto 2D planes. Instead, PointRas takes reference to two core steps in rasterization: **A** determine the pixels covered by a primitive shape; **B** interpolation of the output attributes for all covered pixels. For **A**, we regard the triangle whose vertices are produced by kNN as a primitive shape. Suppose M_{lr} contains N_{lr} points in the current PointRas iteration, then the 3D space is splitted into $\binom{k}{N_{lr}}$ sets of non-overlapping chunks by performing kNN algorithm. Therefore, we can locate the points in M_{hr} which is covered by a certain triangle primitive by querying the points located in this chunk. When $k = 3$, PointRas constructs the triangles in 3D space for interpolation similar to the rasterization rendering in Fig. 2(b). For **B**, the attributes (prediction scores and features) of M_{hr} are interpolated from its vertices of triangle primitive, therefore being similar to rasterization. However, the interpolation targets of rasterization in rendering are defined on the regular pixels which may fall into the empty space, while PointRas interpolates the scattered points on the surface of 3D entity. Therefore, PointRas enjoys the flexibility of target locations. Further optimizations in rasterization rendering such as empty space skipping [64] are not required in PointRas.

4) Regularization Methods v.s. PointRas: Inspired by 2D image segmentation, regularization methods such as CRF [15] and label propagation [16] are proposed for 3D point cloud segmentation as post-processing. The main idea of CRF is to classify the points that are spatially close and geometrically similar into the same label group. While PointRas does not explicitly conduct such post-processing, the kNN and interpolation in Euclidean space naturally makes the features of closer points similar, thereby producing similar semantic predictions after re-prediction using the interpolated features. However, taking CRF in 2D image segmentation as reference, its application in post-processing of 3D point cloud segmentation usually requires the pre-processing of data structure transformation into voxels [65] or range images [25]. Such conversion suffers from extra computation costs and may lose geometry information due to quantization. Moreover, CRF in [65] and [25] is employed solely upon the final up-sampled feature maps without explicitly leveraging the lower-resolution information, while the computation in PointRas is aligned with the network decoders with multiple resolutions. Note that the optimization of CRF is hierarchical (optimize energy function first and loss function second) while PointRas is implemented with differentiable mathematical operations and neural layers which can be optimized fully end-to-end. As for label propagation techniques, the target function minimized in [16] is defined in Equation(6), which shares with CRF the spirit that geometrically similar points are likely to be assigned

TABLE I

NETWORK CONFIGURATION DETAILS FOR THE BACKBONE NETWORKS. SA(N,r,n) DENOTES THE SET ABSTRACTION WHERE N SEED POINTS ARE EXTRACTED USING FURTHEST POINT SAMPLING, AND n NEIGHBOR POINTS ARE QUERIED FOR EACH SEED POINT WITHIN A SPHERE REGION WITH RADIUS r. 3NN(N_1,N_2) INDICATES THE INTERPOLATION FUNCTION IN (2) BASED ON THE WEIGHTED INVERSE DISTANCE FROM N_1 POINTS TO N_2 POINTS, WITH $k = 3$. DE(σ) DENOTES THE DENSITY ESTIMATION FUNCTION IN [1] WITH THE BANDWIDTH OF σ. RES_STRIDED(r,C) INDICATES THE RESNET MODULE WITH STRIDE 2, WHERE r REFERS TO THE GRID LENGTH IN GRID DOWNSAMPLING AND THE DIMENSION OF INPUT/OUTPUT FEATURE CHANNELS ARE $\frac{C}{2}$ AND C RESPECTIVELY. RES(C) DENOTES THE RESNET MODULE WITH IDENTICAL INPUT/OUTPUT FEATURE CHANNELS AND POINT RESOLUTIONS, WHERE C IS THE DIMENSION OF FEATURE CHANNEL. NN DENOTES THE NEAREST NEIGHBOR INTERPOLATION, WHOSE INPUT/OUTPUT POINT RESOLUTIONS ARE PREDEFINED DURING THE PROCESS OF GRID DOWNSAMPLING. MLP(D) INDICATES THE MLP WITH THE SPECIFIED FEATURE DIMENSIONS. MLP_GAB(D) DENOTES THE MLP FOLLOWED BY THE GRAPH ATTENTION BLOCK (GAB) IN [48]

	Encoder1	Encoder2	Encoder3	Encoder4	Decoder4	Decoder3	Decoder2	Decoder1

PointNet++	SA(1024, 0.1, 32)
	MLP(32,32,64)
		SA(256, 0.2, 64)
		MLP(64,64,128)
			SA(64, 0.4, 32)
			MLP(128,128,256)
				SA(16, 0.8, 32)
				MLP(256,256,512)
					3NN(16, 64)
					MLP(256,256)
						3NN(64, 256)
						MLP(256,256)
							3NN(256, 1024)
							MLP(256,128)
								3NN(1024, 8192)
								MLP(128,128,128)

PointConv	SA(1024, 0.1, 32)
	DE(0.05)
	MLP(32,32,64)
		SA(256, 0.2, 32)
		DE(0.1)
		MLP(64,64,128)
			SA(64, 0.4, 32)
			DE(0.2)
			MLP(128,128,256)
				SA(36, 0.8, 32)
				DE(0.4)
				MLP(256,256,512)
					3NN(36,64)
					DE(0.4)
					MLP(512,512)
						3NN(64,256)
						DE(0.2)
						MLP(256,256)
							3NN(256,1024)
							DE(0.1)
							MLP(256,128)
								3NN(1024,8192)
								DE(0.05)
								MLP(128,128,128)

KPConv	Res_Strided(0.08, 128)
	Res(128)
		Res_Strided(0.08, 256)
		Res(256)
			Res_Strided(0.08, 512)
			Res(512)
				Res_Strided(0.08, 1024)
				Res(1024)
					NN
					MLP(1024, 512)
						NN
						MLP(512, 256)
							NN
							MLP(256, 128)
								NN
								MLP(128, 64)

3D-BoNet	SA(1024, 0.1, 32)
	MLP(32,32,64)
		SA(256, 0.2, 64)
		MLP(64,64,128)
			SA(64, 0.4, 128)
			MLP(128,128,256)
				–
				MLP(256,256,512)
					3NN(16, 64)
					MLP(256,256)
						3NN(64, 256)
						MLP(256,256)
							3NN(256, 1024)
							MLP(256,128)
								3NN(1024, 4096)
								MLP(128,128,128,128)

ELGS	SA(1024, 0.1, 32)
	MLP_GAB(32,32,64)
	MLP(64,64)
		SA(256, 0.2, 32)
		MLP_GAB(64,64,128)
		–
			SA(64, 0.4, 32)
			MLP_GAB(128,128,256)
			–
				SA(16, 0.8, 32)
				MLP_GAB(256,256,512)
				–
					3NN(16, 64)
					MLP(256,256)
					–
						3NN(64, 256)
						MLP(256,256)
						–
							3NN(256, 1024)
							MLP(256,128)
							–
								3NN(1024, 4096)
								MLP(128,128,128)
								–

BGA-PN++	SA(512, 0.2, 64)
	MLP(64,64,128)
		SA(128, 0.4, 64)
		MLP(128,128,256)
			–
			MLP(256,512,1024)
				–
				–
					–
					–
						–
						MLP(256,256)
							3NN(128,512)
							MLP(256,128)
								3NN(512,1024)
								MLP(128,128,128)

with the same label.

$$\min_{\{\hat{\boldsymbol{m}}\}} \sum_i \sum_j w_{ij} ||\hat{\boldsymbol{m}}_i - \hat{\boldsymbol{m}}_j||_2^2 + \gamma \sum_i ||\hat{\boldsymbol{m}}_i - \boldsymbol{m}_i||_2^2 \qquad (6)$$

where $\boldsymbol{m}_i$ and $\hat{\boldsymbol{m}}_i$ denote the predicted logit before and after refinement. w_{ij} roughly measures the pairwise similarity in terms of spatial or color manifold. Such strategy shows its effectiveness when tackling the information deficiency of supervision with incomplete labels, where the labeled points are dominantly informative compared with unlabeled ones. In our case, we do not assume such extreme imbalance between lower-resolution and higher-resolution point clouds. Instead, we used the information provided by the lower-resolution point clouds as supplement to the higher-resolution ones.

IV. EXPERIMENT

In this section, we firstly conducted experiments on widely used indoor segmentation datasets ScanNet [2] and S3DIS [20]. By integrating our model into several state-of-the-art backbone networks for 3D point cloud segmentation networks, we get notable improvements upon the baseline methods. We also tested PointRas on the **NPM3D** [21] and **STPLS3D** [22] datasets for outdoor scenes and the **ScanObjectNN** [23] dataset for individual objects with background noises. In addition, we did the ablation studies on several hyper-parameters and key components of our PointRas module and analyze the experimental results.

A. Datasets and Settings

The **ScanNet** v1 [2] dataset contains 1513 indoor scenes with 21 categories. The scans are stored as 3D meshes and vertices, labeled with coordinates and RGB. In our experiments, only point coordinates were used as the input of segmentation networks. Following [6], we sampled 1.5m × 1.5m × 3m blocks, each containing 8,192 points. We followed [2] to split 1201 scenes for training and 312 for testing.

The **S3DIS** [20] dataset contains the images and point clouds from 271 rooms, 6 large-scale indoor areas and 3 buildings. Each point is annotated with one of the 13 semantic labels. 6-fold cross validation (CV) and evaluation on single Area 5 were adopted to test the generalization performance. In 6-fold CV, we followed [10] to compute average mIoU over 6 separated test sets (Areas). We followed [6] to uniformly sample the points into blocks of size 1m × 1m. In each block, 4096 points were randomly sampled on the fly during training and all the points were adopted for testing.

The **NPM3D** dataset [21] contains 2km of lidar point clouds collected by Velodyne HDL-32e in 2 different cities. It contains 160 million points annotated with 10 semantic classes. We evaluated our method on the 9 coarse but representative categories and submitted the prediction results to the official benchmark website.

The real-world split of the **STPLS3D** dataset [22] contains point clouds reconstructed by aerial images from 4 real-world urban sites. Following the official setting, we trained on University of Southern California Park Campus (USC), Orange County Convention Center (OCCC) and a residential area (RA). The evaluation was done on Wrigley Marine Science Center (WMSC)

The **ScanObjectNN** dataset [23] contains 15 categories with around 15,000 indoor objects and 2902 instances collected from the ScanNet [2] and SceneNN [66] dataset. Following the settings in dataset paper, we tested our method on the hardest PB_T50_RS split with background noises. The segmentation aims at filtering out the background points and enable the classifier the awareness of foreground objects.

PointNet++ [6], PointConv [1], BA-GEM [62], 3D-BoNet [63], ELGS [48] and KPConv [10] are adopted as our baseline networks. PointNet++, PointConv, BA-GEM and KPConv comprise the hierarchical architecture with

TABLE II

EXPERIMENTAL RESULTS ON THE SEMANTIC SEGMENTATION OF THE SCANNET DATASET, EVALUATED WITH INTERSECTION OVER UNION (IOU) OVER 20 SEMANTIC CATEGORIES AND THEIR AVERAGE RESULT (MIOU). (*Better Performance in Bold*). *NUMBERS REPORTED IN THE ORIGINAL PAPERS

Method	mIoU	wall	floor	cab	bed	chair	sofa	table	door	window	bkshf	pic	cntr	desk	curt	fridg	show	toil	sink	bath	other
PN++ [6]	58.1	73.1	94.2	44.9	69.7	82.9	68.3	62.2	37.4	42.3	68.3	12.9	56.2	47.7	60.3	34.3	55.5	83.3	56.3	76.3	36.8
PN++$_{Ras}$(Ours)	**59.1**(+1.0)	73.4	94.0	45.3	70.3	83.5	71.3	64.0	38.7	43.3	71.6	13.4	52.6	52.9	58.9	34.1	56.4	84.1	57.8	76.8	38.1
P-Conv* [1]	55.6	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
P-Conv [1]	56.2	73.2	94.0	47.0	61.0	79.4	62.5	63.5	30.9	42.7	71.3	8.6	53.3	50.9	58.6	33.1	43.2	82.8	55.8	74.7	38.3
P-Conv$_{Ras}$(Ours)	**59.2**(+3.0)	72.9	94.9	50.0	69.7	82.2	70.2	65.4	37.5	46.9	68.9	11.8	55.1	52.5	61.2	33.9	51.3	83.5	59.7	79.3	37.9
BA-GEM* [62]	63.4	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
BA-GEM [62]	62.3	78.5	95.1	53.2	74.5	83.1	66.4	66.7	48.0	52.0	75.9	24.8	58.0	55.8	61.0	41.5	50.4	83.9	58.2	78.9	40.5
BA-GEM$_{Ras}$(Ours)	**64.1**(+1.9)	79.2	94.9	58.7	73.0	84.6	68.1	66.9	50.1	52.7	76.3	25.2	57.5	58.3	63.9	45.1	56.2	86.0	58.2	82.3	45.2

TABLE III

EXPERIMENTAL RESULTS ON SEMANTIC SEGMENTATION OF THE SCANNET DATASET. (*Better Performance in Bold*)

Method	OA	VOA
PN++ [6]	82.0	83.2
+*Ras*(Ours)	**83.0**(+1.0)	**84.1**(+0.9)
P-Conv [1]	81.9	83.1
+*Ras*(Ours)	**83.1**(+1.2)	**84.2**(+1.1)
BA-GEM [62]	84.8	85.9
+*Ras*(Ours)	**86.0**(+1.2)	**87.1**(+1.2)

TABLE IV

EXPERIMENTAL RESULTS OF INSTANCE SEGMENTATION ON AREA 5 AND ALL 6 FOLDS OF THE S3DIS DATASET. WE COMPARE OUR METHOD WITH DISCRETIZATION-BASED METHOD [30] AND OTHER POINT-BASED METHODS (LOWER ONES). OUR POINTRAS MODULE IS BUILT ON 3D-BONET [63]. *WE GOT THE EXPERIMENTAL RESULTS FROM PUBLICLY AVAILABLE SOURCE CODE. (*Best Performance in Bold*)

Method	mCov	mWCov	mCov	mWCov
	Area 5		All 6 Folds	
DyCo3D'21 [30]	**63.5**	**64.6**	–	–
SGPN'18 [67]	32.7	35.5	37.9	40.8
ASIS'19 [68]	44.6	47.8	51.2	55.1
3D-BoNet'19* [63]	51.3	53.4	58.3	60.4
IAM'20 [69]	49.9	53.2	54.5	58.0
MPNet'20 [70]	50.1	53.2	55.8	59.7
SP-InsSem'20 [71]	54.1	56.3	60.4	**63.0**
3D-BoNet$_{Ras}$(Ours)	53.9	55.1	**61.1**	62.5

downsampling and upsampling layers to achieve different resolutions. 3D-BoNet uses a backbone network to extract the point-wise and global feature vectors, followed by 2 branches of instance-level box prediction and point-level instance segmentation. ELGS uses an encoder-decoder architecture backbone similar to PointNet++ to learn semantic features, where the input features are enhanced by a point enrichment layer. BA-GEM utilizes the supervision on boundary features at early stage of encoder and late stage of decoder, which does not affect the incorporation of the proposed PointRas module.

B. ScanNet Segmentation

The **single-scale grouping** (SSG) version of PointNet++ [6], PointConv [1] (P-Conv) and BA-GEM [62] were used as baseline methods on ScanNet semantic segmentation. The resolution upscaling of the incorporated PointRas is $\times\frac{1}{128}\rightarrow\times\frac{1}{32}\rightarrow\times\frac{1}{8}\rightarrow\times1$ (T = 3). We tested out method on the metric of intersection over union (IoU) of 20 semantic categories and their average (mIoU). We also evaluated the predictions on whole scene on through point-based overall accuracy (OA) and voxel-based overall accuracy (VOA). All metrics are the higher the better. We summarize the experimental performances of mIoU and per-category IoU in TABLE II, point-based and voxel-based OAs in TABLE III. We can see that PointRas module improves OA, VOA and mIoU of both PointNet++ and PointConv network. Notably, the equipment of PointRas module significantly improves the mIoU of the whole scene by 2.7 over the PointConv baseline.

C. S3DIS Segmentation

3D-BoNet, ELGS and KPConv were selected as the baseline network for S3DIS segmentation tasks. We directly incorporated the PointRas module into upsampling layer of KPConv, and did the same thing to the backbone of 3D-BoNet and ELGS. As a default setting, the number of points in initial prediction scores were all set as 64. By default, we adopt the same interpolation method with the baseline method. The resolution upscaling for the S3DIS dataset is $\times\frac{1}{64}\rightarrow\times\frac{1}{16}\rightarrow\times\frac{1}{4}\rightarrow\times1$ (T = 3). For instance segmentation, the evaluation metrics are mean average instance-wise IoU (mCov) and its weighted results (mWCov) [68].

The experimental results of 3D instance segmentation are summarized in TABLE IV. We categorized the compared methods into discretization-based [30] and other point-based methods. The discretization-based DyCo3D outperforms all point-based methods on the evaluation of Area 5. Based on the earlier backbone of 3D-BoNet [63], PointRas significantly improves mCov and mWCov on both Area 5 and 6-fold validation and achieves comparable results with recent point-based methods. Notably, 3D-BoNet equipped with PointRas achieves state-of-the-art result on mCov of 6-fold validation against the other methods.

For semantic segmentation, we incorporate PointRas into KPConv and ELGS. The evaluation metrics are ovarall accuracy (OA), mean accuracy over all categories (mAcc) and mean IoU over all categories (mIoU). The results of Area 5 and 6-fold CV are shown in TABLE V. Note that the recent methods [31], [40] based on RandLANet [39] adopts another

TABLE V

Experimental Results of Semantic Segmentation on Area 5 and All 6 Folds of the S3DIS Dataset. Our PointRas Module Is Built on ELGS [48] and KPConv-*rigid* [10]. *: We Re-Organized the Results of SCF-Net [40] and BAAF-Net [31] Following the Cross-Validation Protocol in [10]. (*Best Performance in Bold*)

	Method	OA	mAcc	mIoU	ceiling	floor	wall	beam	column	window	door	chair	table	bookcase	sofa	board	clutter
Area 5	Pointnet'17 [5]	–	49.0	41.1	88.8	97.3	69.8	**0.1**	3.9	46.3	10.8	58.9	52.6	5.9	40.3	26.4	33.2
	SPGraph'18 [15]	86.4	66.5	58.0	89.4	96.9	78.1	0.0	**42.8**	48.9	61.6	75.4	84.7	52.6	69.8	2.1	52.2
	PointCNN'18 [9]	85.9	63.9	57.3	92.3	98.2	79.4	0.0	17.6	22.8	62.1	74.4	80.6	31.7	66.7	62.1	56.7
	ELGS'19 [48]	88.4	–	60.1	92.8	98.5	72.7	0.0	32.4	68.1	28.8	74.9	85.1	55.9	64.9	47.7	58.2
	KPConv'19 [10]	–	70.9	65.4	92.6	97.3	81.4	0.0	16.5	54.5	69.5	90.1	80.2	74.6	66.4	63.7	58.1
	SPH3D-GCN'20 [72]	87.7	65.9	59.5	93.3	97.1	81.1	0.0	33.2	45.8	43.8	79.7	86.9	33.2	71.5	54.1	53.7
	SCF-Net'21 [40]	–	–	63.4	–	–	–	–	–	–	–	–	–	–	–	–	–
	BAAF-Net'21 [31]	88.9	73.1	65.4	92.9	97.9	82.3	0.0	23.1	65.5	64.9	78.5	87.5	61.4	70.7	**68.7**	57.2
	PAConv'21 [32]	–	73.0	66.6	**94.6**	**98.6**	82.4	0.0	26.4	58.0	60.0	89.7	80.4	74.3	69.8	73.5	57.7
	ELGS$_{Ras}$(Ours)	88.5	66.6	62.6	92.8	97.3	73.4	0.0	18.7	**68.4**	50.3	77.3	85.9	63.8	68.0	60.8	57.3
	KPConv$_{Ras}$(Ours)	**90.0**	**73.3**	**67.4**	94.5	98.2	**83.9**	0.0	24.8	57.8	64.1	**91.6**	81.8	**75.3**	**77.2**	66.7	**60.0**
All 6 Folds	Pointnet'17 [5]	78.5	66.2	47.6	88.0	88.7	69.3	42.4	23.1	47.5	51.6	42.0	54.1	38.2	9.6	29.4	35.2
	SPGraph'18 [15]	85.5	73.0	62.1	89.9	95.1	76.4	62.8	47.1	55.3	68.4	73.5	69.2	63.2	45.9	8.7	52.9
	PointCNN'18 [9]	88.1	75.6	65.4	**94.8**	**97.3**	75.8	**63.3**	51.7	58.4	57.2	71.6	69.1	39.1	61.2	52.2	58.6
	ELGS'19 [48]	87.6	–	66.3	93.7	95.6	76.9	42.6	46.7	63.9	69.0	70.1	76.0	52.8	57.2	54.8	62.5
	KPConv'19 [10]	–	78.1	69.6	93.7	92.0	82.5	62.5	49.5	65.7	77.3	57.8	64.0	68.8	71.7	60.1	59.6
	SPH3D-GCN'20 [72]	88.6	77.9	68.9	93.3	96.2	81.9	58.6	**55.9**	55.9	71.7	72.1	**82.4**	48.5	64.5	54.8	60.4
	PAConv'21 [32]	–	78.7	69.3	94.3	93.5	82.8	56.9	45.7	65.2	74.9	59.7	74.6	67.4	61.8	**65.8**	58.4
	SCF-Net'21 [40]*	–	–	69.5	–	–	–	–	–	–	–	–	–	–	–	–	–
	BAAF-Net'21 [31]*	**89.2**	**81.9**	70.9	93.9	96.9	81.8	48.8	47.6	64.4	76.4	69.0	83.7	**67.7**	63.3	64.7	**64.4**
	ELGS$_{Ras}$(Ours)	87.8	73.4	67.4	93.0	95.2	77.5	50.4	37.0	63.2	74.4	70.3	76.5	57.1	58.9	59.8	62.0
	KPConv$_{Ras}$(Ours)	88.9	79.3	**71.4**	94.6	93.7	**83.9**	45.3	53.9	**69.6**	**79.4**	**73.6**	65.8	67.4	**76.2**	60.5	63.9

k-fold CV strategy, which traverses testing results over all samples in 6 Areas and computes overall mIoU. For fair comparison, we re-evaluated the segmentation results following the averaging strategy of 6-fold CV in previous works [10]. We can see that KPConv equipped with PointRas achieves state-of-the-art performance on all 3 metrics of Area 5. It also outperforms the other methods in terms of mIoU in 6-fold CV, and is comparable with state-of-the-art BAAF-Net [31] in terms of OA and mAcc. This verifies that PointRas enhances the representation ability of KPConv-*rigid* baseline.

As for ELGS [48] baseline, PointRas improves the performance on mIoU (2.5↑ for Area 5 and 1.1↑ for 6-fold CV). Notably, ELGS itself contains a well-designed prediction layer comprised of both channel and spatial attention to get the final $M_{backbone}$. In consideration of consistency with other backbones and computational efficiency, we direct use the output of PointRas to partially replace $M_{backbone}$, without going through those attention mechanisms again for PointRas.

For individual categories, ELGS and KPConv get obvious improvements or at least competitive results on IoU performances. Notably, taking Area 5 of S3DIS as example, the segmentation performance of ELGS on 'door' category improves by 21.5%. And categories of 'ceiling', 'floor', 'wall', 'window' and 'board' consistently gain obvious IoU boost after incorporating PointRas into KPConv. Note that those categories of objects are relatively plain in terms of geometric features, which verifies the hypothesis that PointRas can well depict them in lower resolutions.

D. ScanObjectNN, NPM3D and STPLS3D Segmentation

TABLE VI

Comparisons of Simultaneous Foreground Segmentation and Object Classification on the ScanObjectNN [23] Dataset, and Semantic Segmentation Performed on the NPM3D [21] and STPLS3D [22] Datasets

Method	ScanObjectNN			NPM3D	STPLS3D
	OA$_{seg}$	OA	mAcc	mIoU	
PointNet'17 [5]	–	68.2	63.4	–	–
PointNet++'17 [6]	–	77.9	75.4	–	–
SpiderCNN'18 [73]	–	73.7	69.8	–	–
DGCNN'19 [34]	–	78.1	73.6	–	–
PointCNN'18 [9]	–	78.5	75.1	–	–
KPConv'19 [10]	–	79.7	77.5	75.9	45.2
BGA-DGCNN'19 [23]	–	79.7	75.7	–	–
BGA-PN++'19 [23]	77.3	80.4	77.5	–	–
RandLANet'19 [39]	–	–	–	–	42.3
DRNet'21 [56]	–	80.3	**78.0**	–	–
SCF-Net'21 [40]	–	–	–	–	45.9
Ours	on BGA-PN++			on KPConv	
	78.4	**81.2**	**78.0**	**78.4**	**47.4**

To test the generalization of PointRas to individual objects with noises, we also tested PointRas via simultaneous object classification and foreground segmentation on the **ScanObjectNN** [23] dataset. We report the overall foreground segmentation accuracy (OA$_{seg}$, overall classification accuracy (OA) and the mean classification accuracy of 15 categories (mAcc). We adopted BGA-PN++ [23] network as the baseline backbone and equipped its segmentation branch with our PointRas module. To validate whether PointRas can achieve satisfactory performance gains in outdoor datasets, we then validated PointRas on the **NPM3D** [21] and **STPLS3D** [22] datasets. We chose to incorporate PointRas into KPConv-*rigid* [10] network. As the author did not release the detailed network architectures and configurations of KPConv on NPM3D, we kept the architectures and configurations of KPConv the same with KPConv-*rigid* on S3DIS. We report mean IoU (mIoU) of 9 categories as evaluation metric.

The experimental results are shown in TABLE VI. We can see that the performance of foreground segmentation is enhanced by an obvious margin (OA$_{seg}$ ↑ 1.1). Moreover, the

TABLE VII
EXPERIMENTAL RESULTS ON SEGMENTATION AND CLASSIFICATION OF THE SCANOBJECTNN DATASET. (*Best Performance in Bold*)

Method	OA_{seg}	OA	mAcc	bag	bin	box	cabinet	chair	desk	display	door	shelf	table	bed	pillow	sink	sofa	toilet
PointNet'17 [5]	–	68.2	63.4	36.1	69.8	10.5	62.6	89.0	50.0	73.0	93.8	72.6	67.8	61.8	67.6	64.2	76.7	55.3
PointNet++'17[6]	–	77.9	75.4	49.4	84.4	31.6	77.4	91.3	74	79.4	85.2	72.6	72.6	75.5	81.0	80.8	90.5	85.9
SpiderCNN'18 [73]	–	73.7	69.8	43.4	75.9	12.8	74.2	89.0	65.3	74.5	91.4	78.0	65.9	69.1	80.0	65.8	90.5	70.6
PointCNN'18 [9]	–	78.5	75.1	57.8	82.9	33.1	83.6	**92.6**	65.3	78.4	84.8	**84.2**	67.4	80.0	80.0	72.5	91.9	71.8
DGCNN'19 [34]	–	78.1	73.6	49.4	82.4	33.1	83.9	91.8	63.3	77.0	89.0	79.3	**77.4**	64.5	77.1	75	91.4	69.4
DRNet'21 [56]	–	80.3	**78.0**	**66.3**	81.9	**49.6**	76.3	91.0	65.3	**92.2**	91.4	83.8	71.5	**79.1**	75.2	75.8	91.9	78.8
PRA-Net'21 [49]	–	81.0	77.9	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
BGA-DGCNN'19 [23]	–	79.7	75.7	48.2	81.9	30.1	**84.4**	**92.6**	**77.3**	80.4	**92.4**	80.5	74.1	72.7	78.1	**79.2**	91.0	72.9
BGA-PN++'19 [23]	77.3	80.4	77.5	54.2	85.9	39.8	81.7	90.8	76.0	84.3	87.6	78.4	74.4	73.6	**80.0**	77.5	91.9	**85.9**
BGA-PN++$_{Ras}$(Ours)	**78.4**	**81.2**	**78.0**	56.6	**86.4**	45.9	80.4	92.1	74.7	83.8	91.0	80.5	71.5	75.5	75.2	**79.2**	**93.3**	84.7

TABLE VIII
THE ABLATION STUDIES ON THE TO VERIFY WHETHER INDIVIDUAL COMPONENT IS USEFUL. (*Best Performance in Bold*)

Re-prediction	Uncertainty-aware	PointNet++		PointConv		KPConv	
		OA	mIoU	OA	mIoU	OA	mIoU
		82.0	58.1	81.9	56.2	–	65.4
✓		82.7	59.0	82.3	57.1	89.6	66.3
✓	✓	**83.0**	**59.1**	**82.7**	**58.7**	**90.0**	**67.4**

TABLE IX
THE ABLATION STUDIES ON THE INITIAL RESOLUTION OF THE POINTRAS MODULE, I.E., THE NUMBER OF ITERATIONS WITHIN THE POINTRAS MODULE. (*Best Performance in Bold*)

Initial Resolution	PointNet++		ELGS	
	OA	mIoU	OA	mIoU
baseline	82.0	58.1	88.1	61.4
(1) start from $\boldsymbol{l}_4$	81.8	56.2	87.9	61.0
(2) start from $\boldsymbol{l}_2$	81.0	57.0	87.6	60.4
start from $\boldsymbol{l}_3$	**83.0**	**59.1**	**88.5**	**62.6**

classification branch also benefits from the training process of PointRas (OA ↑ 0.8, mAcc ↑ 0.5). As for performances of individual categories, we thoroughly display their classification accuracy in TABLE VII. For NPM3D and real-world split of STPLS3D, compared to KPConv baseline, PointRas improves mIoU by 2.5% and 2.5% respectively, which verifies the effectiveness of PointRas in outdoor scenes.

E. Ablation Study

We further conducted thorough ablation studies on the setting of hyper-parameters and the key components of the PointRas module.

1) Key Components: To validate whether individual components are useful, we have conducted the corresponding ablation studies and the experimental results are shown in Table VIII. We either removed the final re-prediction entirely, or replaced the uncertainty-aware selection with random selection. The results in Table VIII demonstrates the effects of the key components intuitively, where both the uncertainty-aware selection and final re-prediction help to improve the segmentation performances. The experiments of PointNet++ and PointConv were done on ScanNet. The experiments of KPConv were done on S3DIS.

2) Initial Resolution: We also modified the initial resolution where M_{lr} is initialized in PointRas. The backbone networks are PointNet++ on the ScanNet and ELGS on the S3DIS, with 4 feature propagation layers ($\boldsymbol{l}_4$-$\boldsymbol{l}_3$-$\boldsymbol{l}_2$-$\boldsymbol{l}_1$-$\boldsymbol{l}_0$) in an ascending order of resolution. Specifically, we modified the source of M_{lr} from the default $\boldsymbol{l}_3$ (64 points, $T = 3$) to $\boldsymbol{l}_4$ (lower resolution, 16 points, $T = 4$) or $\boldsymbol{l}_2$ (higher resolution, 256 points, $T = 2$). From the experimental results in TABLE IX, we can seen that the segmentation performances fall behind with lower or higher initial resolution compared to default setting. We infer that compared to the default setting, the information deficiency is overwhelming given only a quarter of the initial points, while the spatial redundancy increases with 4× initial resolution.

3) Proportion of Refined Points: We then changed the proportion of the refined points in each iteration of the PointRas module (α), and the proportion of the replaced points for the backbone predictions after the last iteration (β). The backbone networks are PointNet++ on the ScanNet and KPConv on the S3DIS. As shown in TABLE X, we can verify that choosing a proper proportion ($\alpha = 1/2$) of uncertain points to re-predict is better than re-predicting none or all of the points. During the selection of prediction with uncertainty for the final resolution, the default ratio of replacement ($\beta = 1/6$) appropriately controls the extent.

While the default setting achieves the best performance, setting (1) in TABLE X also achieves the superior performance over baseline in terms of OA and mIoU. In other words, after training, if we decouple the PointRas module from the backbone network, the segmentation performance still improves upon the baseline. This implies that the representation ability in lower resolution of **backbone** network is also enhanced with the training of PointRas module, where the optimizing target in (5) can be regarded as a variant of the *companion loss* [74]. However, the default setting or a small modification on β still achieves the best performance in both metrics, which verifies the necessity of the proposed architecture.

4) Uncertainty Criterion and Interpolation: We have also explored the nearest neighbor (NN, $k = 1$) and the inverse distance weighted (3NN, $k = 3$) method for the feature and score interpolation in Fig. 3. Moreover, we compared the performance of uncertainty selection in the criterion of

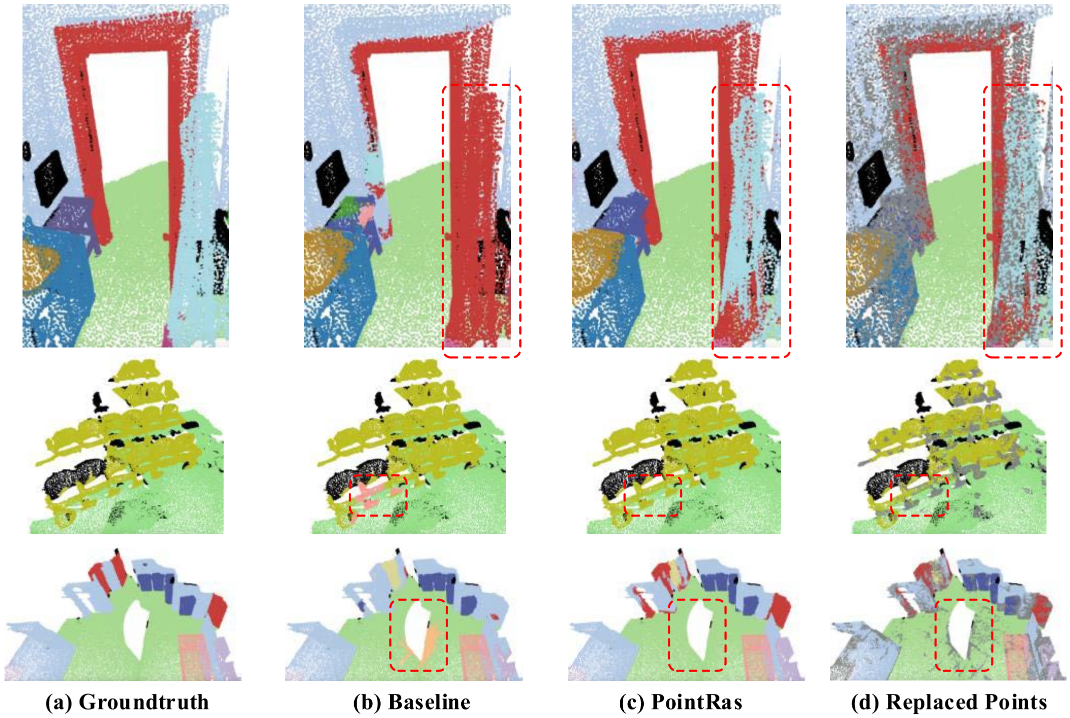

Fig. 4. Qualitative results from the predictions with the backbone of PointConv [1]. **(a)** The groundtruth labels of the whole scene. **(b)** The segmentation results predicted by the baseline method. **(c)** The segmentation results predicted by the baseline method incorporated with our proposed PointRas module. **(d)** Based on (c), we present the replaced points after the last iteration of PointRas module in gray.

TABLE X

THE ABLATION STUDIES ON THE PROPORTION OF REFINED POINTS WITHIN THE ITERATION OF POINTRAS MODULE (α), AND AFTER THE LAST POINTRAS ITERATION (β). (*Best Performance in Bold*)

Replacement Ratio	PointNet++		KPConv	
	OA	mIoU	OA	mIoU
baseline	82.0	58.1	–	65.4
(1) $\beta = 0$ (w/o replace)	82.7	59.0	89.6	66.3
(2) $\alpha = 0, \quad \beta = 1/6$	82.5	58.7	89.6	65.6
(3) $\alpha = 1, \quad \beta = 1/6$	82.7	58.6	89.8	66.6
(4) $\alpha = 1/2, \beta = 1/8$	82.8	58.8	**90.0**	**67.6**
(5) $\alpha = 0, \quad \beta = 1/4$	82.3	58.2	89.2	65.0
(6) $\alpha = 1/2, \beta = 1$	82.4	57.8	89.8	66.5
$\alpha = 1/2, \beta = 1/6$	**83.0**	**59.1**	**90.0**	67.4

TABLE XI

THE ABLATION STUDIES ON THE METHOD OF INTERPOLATION AND CRITERION OF UNCERTAINTY. THE CHECKMARKS IN GREEN AND RED DENOTE THE DEFAULT SETTINGS FOR POINTCONV AND KPCONV BACKBONE RESPECTIVELY. (*Best Performance in Bold*)

Feat. Interp.		Score Interp.		Uncertainty		PointConv		KPConv	
NN	3NN	NN	3NN	Self	Comp.	OA	mIoU	OA	mIoU
Baseline						81.9	56.2	-	65.4
✓		✓		✓		83.0	**59.1**	**90.0**	**67.4**
	✓	✓		✓		83.0	59.0	89.7	66.3
✓			✓	✓		83.0	**59.1**	89.7	66.2
	✓		✓	✓		**83.1**	58.9	89.4	66.2
✓		✓			✓	82.7	58.8	89.7	66.6
	✓		✓		✓	82.7	58.7	89.8	66.5

Self Digging and **Comparison** (Comp.). The backbone networks are PointConv on the ScanNet and KPConv-*rigid* on the S3DIS. From the results in TABLE XI, we can see that the equipped PointRas in PointConv outperforms the baseline method by an obvious margin under various settings. While it is intuitive that a larger k enhances the representation ability when upscaling, keeping the interpolation method in PointRas consistent with the backbone network is more suitable for KPConv. Moreover, the uncertainty criterion based on self digging analyzes the fine-grained category scores, while the norm-based comparing method loses the ranking information of categories. Note that PointRas still outperforms the baseline method after various modifications, which validates enhancement of representation ability endowed by the proposed module.

Instead of verifying whether the usage of a proposed component takes effect in TABLE VIII, the ablation studies in TABLE XI aims at verifying whether the proposed component is sensitive when implemented by other alternative ways. Overall, the combinations in TABLE XI have yielded small variation, which shows that the alternative implementations cannot outperform the default implementation in PointRas.

5) Distance Metrics: For the computation in Equation (2), we use L2 distance by default since only 3-dimensional coordinates are considered in PointRas. To further investigate the selection of distance metric, we compared our default setting with (1)-cosine similarity between 3-dimensional coordinates for both F_j^{hr} and $\hat{M}_j^{hr}$; (2)-L2 distance between 3-dimensional coordinates for F_j^{hr}, cosine similarity between higher-dimensional features for $\hat{M}_j^{hr}$; (3)-cosine similarity between 3-dimensional coordinates for F_j^{hr}, cosine distance between higher-dimensional features for $\hat{M}_j^{hr}$. Note that for F_j^{hr}, the weights cannot be generated from high-dimensional

TABLE XII

EXPERIMENTAL RESULTS OF SEMANTIC SEGMENTATION ON AREA 5 OF THE S3DIS DATASET. EXPERIMENTS ARE DONE ON ELGS [48]. BY DEFAULT, WE USE L2 DISTANCE BETWEEN COORDINATES IN EUCLIDEAN SPACE. HERE WE HAVE FURTHER COMPARED IT WITH ANOTHER 3 SETTINGS. (*Best Performance in Bold*)

Method	OA	mAcc	mIoU
ELGS [48]	88.4	–	60.1
PointRas	**88.5**	**66.6**	**62.6**
Metric (1)	82.1	55.6	52.1
Metric (2)	86.4	63.0	59.2
Metric (3)	82.1	55.8	52.1

features. This is because the features in PointRas is computed on-line as opposed to 3D coordinates. The experimental results of ELGS [48] on the Area 5 of the S3DIS datasets are summarized in Table XII. We can observe that cosine similarity in both 3-dimensional Euclidean space and higher-dimensional feature space is harmful to the segmentation performance. The L2 distance in 3-dimensional Euclidean space is both computationally efficient and experimentally effective.

F. Comparison With Regularization-Based Methods

To compare our PointRas with regularization-based methods such as label propagation in [16], we added a discriminator g on ELGS [48] which learns the pairwise weights on geometric manifold $\{w_{ij}|i, j = 1, \ldots, N\}$ for the point-wise features in PointRas. During training, the following regularization function is minimized:

$$l_r = \frac{1}{||\boldsymbol{W}||_0} \sum_i \sum_j w_{ij} ||g(\boldsymbol{F}_i) - g(\boldsymbol{F}_j)||_2^2 \qquad (7)$$

where $\{\boldsymbol{F}_i|i = 1, \ldots, N\}$ are point-wise features generated by PointRas. In practice we use 1-norm for $\boldsymbol{W}$ for traceable gradient computation. During inference, we follow CRF to conduct a hierarchical refinement: we firstly refine the prediction scores produced by the last iteration of PointRas by optimizing Equation(6), then use them to refine the prediction scores produced by the backbone network. The experiments are done on ELGS [48] of the S3DIS dataset. The hyper-parameters are kept the same with default settings. Based on the default settings, we have further conducted 2 trials. Firstly, we strictly followed [16] denoted as “ELGS+PointRas+Reg.[16]”. Secondly, we directly implemented the regularization upon the results of [48] denoted as “ELGS+Reg.[16]”. Following [16], we set γ as 1. The weight for l_r was 5e-2.

The experimental results are summarized in Table XIII, where we report both segmentation performance and inference time per batch (batch size set to 1, 4096 points per sample). We can see that solving the regularization function for the final highest resolution brings 1.5x as much extra inference time as ELGS baseline method (see ELGS+Reg.[16]), while bringing only marginal extra cost in lower resolution (see ELGS+PointRas+Reg.[16]). Moreover, the regularization in PointRas does not improve the overall metrics and notably worsens ELGS if implemented at the highest resolution. We infer that current convolution operators tend to encode individual points with contextual information homogeneously (such as the GPM in ELGS [48]), which may contradict with the implemented regularization [16] that requires discriminative feature representation for every point. Instead, PointRas resorts to heterogeneous feature representation, i.e., in multiple resolutions, which can be informative supplement to the predictions in the highest resolution.

TABLE XIII

EXPERIMENTAL RESULTS OF SEMANTIC SEGMENTATION ON AREA 5 OF THE S3DIS DATASET. EXPERIMENTS ARE DONE ON ELGS [48]. APART FROM THE EXPERIMENTS IN ORIGINAL MANUSCRIPT, WE HAVE FURTHER IMPLEMENTED POINTRAS AND ELGS WITH MANIFOLD REGULARIZER IN [16]. (*Best Performance in Bold*)

Method	Time (ms)	OA	mAcc	mIoU
ELGS [48]	**40.6**	88.4	–	60.1
ELGS+PointRas	44.9	**88.5**	**66.6**	**62.6**
ELGS+Reg.[16]	101.0	87.0	61.7	56.4
ELGS+PointRas+Reg.[16]	46.6	**88.5**	66.4	**62.6**

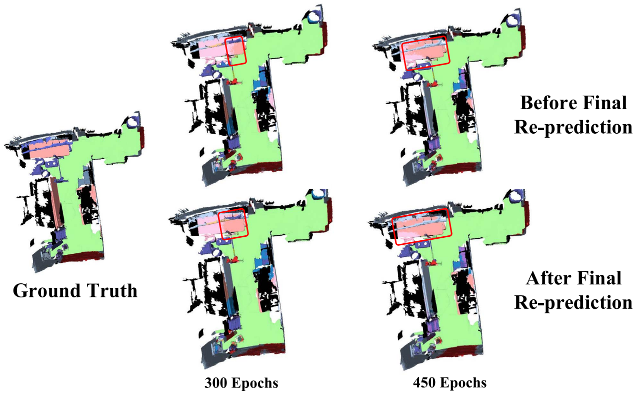

Fig. 5. Visualization results of the predicted semantics by (1) before and after the final re-prediction (2) different training stages (epochs).

G. Computational Cost

For the aforementioned baseline methods, we calculate the ratio of additional FLOPs (multiply-adds) when incorporated with our PointRas module under the default settings. As shown in TABLE XIV, the biggest ratio of additional FLOPs is only 2.45% when evaluating on the PointNet++ equipped with PointRas module. The other ratios when evaluating are no more than 1.13%. This verifies that PointRas module is light-weighted and computationally efficient, while boosting the segmentation performances notably.

H. Qualitative Results

in Figure 5, we have visualized the predicted semantics by (1) before and after the final re-prediction (2) different training stages (epochs). We used PointConv [1] network on ScanNet [2]. We can observe that as the learning progresses, the overall segmentation accuracy is improved. More importantly, as indicated by the red rectangles in Figure 5, the refinement by the final re-prediction is obvious by comparing the visualization results before and after the final re-prediction. We further illustrate 3 samples from the ScanNet dataset in Fig. 4. In Fig. 4(d), we present the updated predictions after the last iteration of PointRas in gray. As the evaluation batches

TABLE XIV

THE RATIO OF ADDITIONAL FLOPS TO THE BASELINE METHOD, WHEN INCORPORATED WITH OUR POINTRAS MODULE. WE CALCULATE THE RATIO AT BOTH THE TRAINING (RAS-*train*) AND EVALUATION (RAS-*eval*) PHASE. $*$ AND $**$ DENOTE THE EXPERIMENTS ON S3DIS AND NPM3D DATASETS RESPECTIVELY

	PN++	P-Conv	3D-BoNet	ELGS	KPConv*	KPConv**	BGA-PN++	BA-GEM
Ras-*train*	+0.12%	+0.03%	+0.04%	+0.03%	+0.29%	+0.22%	+0.07%	+0.04%
Ras-*eval*	+2.45%	+0.61%	+0.36%	+0.22%	+0.35%	+0.30%	+1.13%	+0.43%

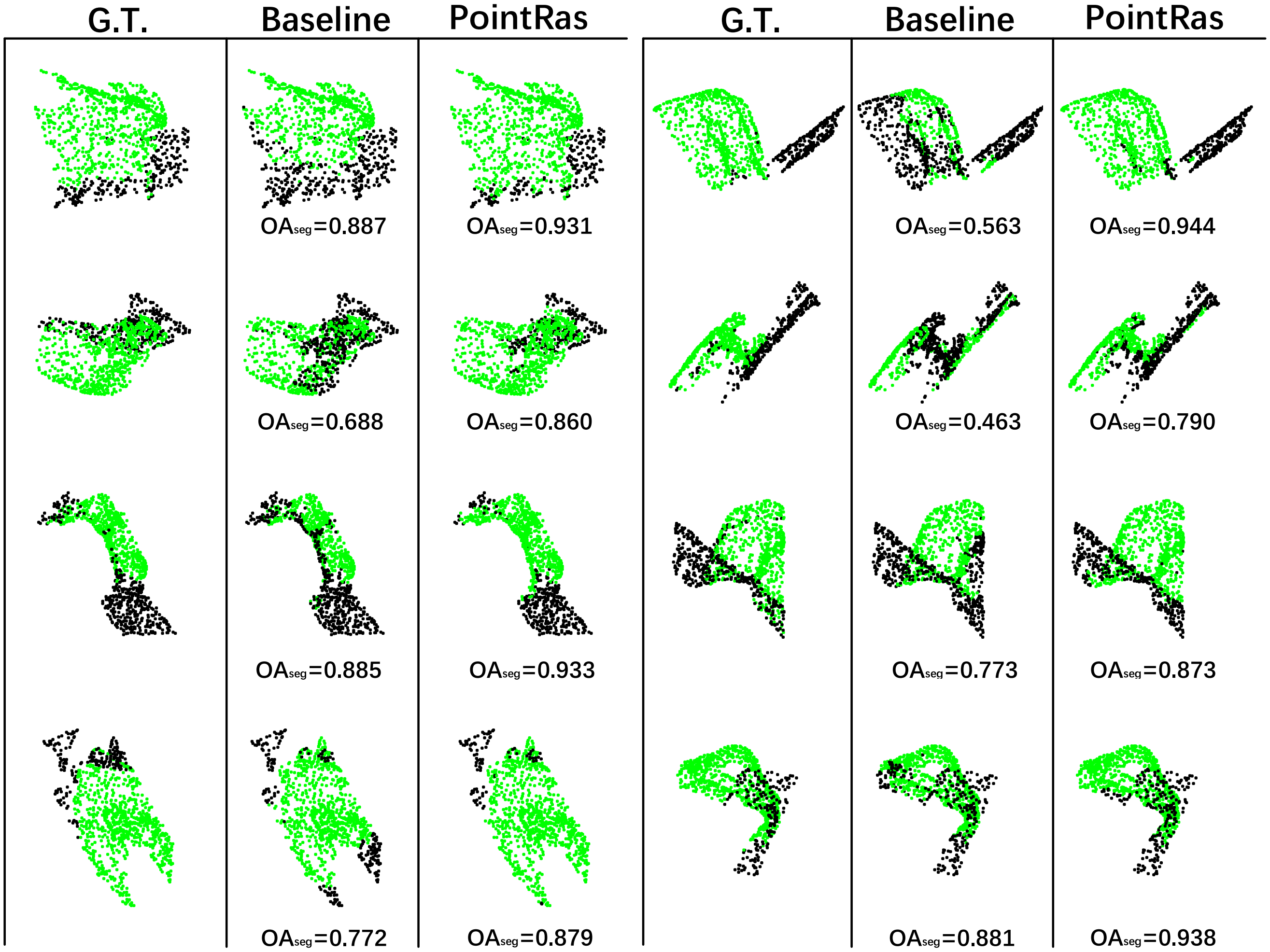

Fig. 6. Qualitative results from the ScanObjectNN dataset. **Green** points and **black** points indicate the **foreground** and **background** points respectively. We illustrate the groundtruth (G.T.) annotations, the predictions of BGA-PN++ baseline and the results after incorporating PointRas module.

intersect a lot, we only illustrate top one-fifth of the replaced points with the largest uncertainty. We have highlighted the distinguishing regions in (b), (c) and (d) by using colored boxes. In those regions, our PointRas achieves significant better segmentation results than baseline method in reference to groundtruth. We observe that the most uncertain predictions appear at the junction of different instances, or the edge of the shape. We have also provided the qualitative results on the foreground segmentation on the ScanObjectNN dataset in Fig. 6. Compared to the ground-truth (G.T.) annotations, PointRas enhances the ability of discrimination between foreground and background points.

V. CONCLUSION AND FUTURE WORK

In this paper, we have proposed an uncertainty-aware multi-resolution learning for point cloud Segmentation, named PointRas. Taking reference to rasterization for 3D rendering, we regress the prediction scores in lower resolutions and interpolate them into higher resolutions, which aims at exploiting their representation ability. Moreover, to remedy the potential information deficiency of lower-resolution points, we refine the prediction scores in the criterion of uncertainty selection. Experimental results on multiple datasets validate the effectiveness and efficiency of our method. In the future work, we intend to research on the fitness of PointRas on other variants of the backbones, such as the voxel-based 3D U-Net for point-wise feature extraction.

REFERENCES

[1] W. Wu, Z. Qi, and L. Fuxin, “PointConv: Deep convolutional networks on 3D point clouds,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9621–9630.

[2] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Niessner, "ScanNet: Richly-annotated 3D reconstructions of indoor scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5828–5839.

[3] B. Graham, M. Engelcke, and L. V. D. Maaten, "3D semantic segmentation with submanifold sparse convolutional networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 9224–9232.

[4] Y. A. Alnaggar, M. Afifi, K. Amer, and M. ElHelw, "Multi projection fusion for real-time semantic segmentation of 3D LiDAR point clouds," in *Proc. WACV*, 2021, pp. 1800–1809.

[5] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 652–660.

[6] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in *Proc. NeurIPS*, 2017, pp. 5099–5108.

[7] Y. Liu, B. Fan, S. Xiang, and C. Pan, "Relation-shape convolutional neural network for point cloud analysis," in *Proc. CVPR*, 2019, pp. 8895–8904.

[8] Y. Duan, Y. Zheng, J. Lu, J. Zhou, and Q. Tian, "Structural relational reasoning of point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 949–958.

[9] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, "PointCNN: Convolution on χ-transformed points," in *Proc. NeurIPS*, 2018, pp. 820–830.

[10] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. Guibas, "KPConv: Flexible and deformable convolution for point clouds," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6411–6420.

[11] A. Komarichev, Z. Zhong, and J. Hua, "A-CNN: Annularly convolutional neural networks on point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7421–7430.

[12] Z. Zhang, B.-S. Hua, and S.-K. Yeung, "ShellNet: Efficient point cloud convolutional neural networks using concentric shells statistics," in *Proc. ICCV*, 2019, pp. 1607–1616.

[13] Q. Zhang, Y. N. Wu, and S.-C. Zhu, "Interpretable convolutional neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8827–8836.

[14] A. Grunnet-Jepsen, J. N. Sweetser, and J. Woodfill, "Best-known-methods for tuning intel realsense d400 depth cameras for best performance," Intel Corp., Satan Clara, CA, USA, Tech. Rep., 2018, vol. 1.

[15] L. Landrieu and M. Simonovsky, "Large-scale point cloud semantic segmentation with superpoint graphs," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4558–4567.

[16] X. Xu and G. H. Lee, "Weakly supervised semantic point cloud segmentation: Towards 10X fewer labels," in *Proc. CVPR*, 2020, pp. 13706–13715.

[17] Z. Yang, Y. Sun, S. Liu, X. Qi, and J. Jia, "CN: Channel normalization for point cloud recognition," in *Proc. ECCV*, 2020, pp. 600–616.

[18] L. Yang, Y. Han, X. Chen, S. Song, J. Dai, and G. Huang, "Resolution adaptive networks for efficient inference," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 2369–2378.

[19] A. Kirillov, Y. Wu, K. He, and R. Girshick, "PointRend: Image segmentation as rendering," in *Proc. CVPR*, 2020, pp. 9799–9808.

[20] I. Armeni *et al.*, "3D semantic parsing of large-scale indoor spaces," in *Proc. ICCV*, 2016, pp. 1534–1543.

[21] X. Roynard, J.-E. Deschaud, and F. Goulette, "Paris-Lille-3D: A large and high-quality ground-truth urban point cloud dataset for automatic segmentation and classification," *Int. J. Robot. Res.*, vol. 37, no. 6, pp. 545–557, 2018.

[22] M. Chen *et al.*, "STPLS3D: A large-scale synthetic and real aerial photogrammetry 3D point cloud dataset," 2022, *arXiv:2203.09065*.

[23] M. A. Uy, Q.-H. Pham, B.-S. Hua, T. Nguyen, and S.-K. Yeung, "Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data," in *Proc. ICCV*, 2019, pp. 1588–1597.

[24] Y. Lyu, X. Huang, and Z. Zhang, "Learning to segment 3D point clouds in 2D image space," in *Proc. CVPR*, 2020, pp. 12255–12264.

[25] B. Wu, A. Wan, X. Yue, and K. Keutzer, "SqueezeSeg: Convolutional neural nets with recurrent CRF for real-time road-object segmentation from 3D LiDAR point cloud," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 1–5.

[26] X. Liu, Z. Han, Y.-S. Liu, and M. Zwicker, "Fine-grained 3D shape classification with hierarchical part-view attention," *IEEE Trans. Image Process.*, vol. 30, pp. 1744–1758, 2021.

[27] C. Choy, J. Gwak, and S. Savarese, "4D spatio-temporal ConvNets: Minkowski convolutional neural networks," in *Proc. CVPR*, Jun. 2019, pp. 3075–3084.

[28] F. Engelmann, M. Bokeloh, A. Fathi, B. Leibe, and M. Nießner, "3D-MPA: Multi-proposal aggregation for 3D semantic instance segmentation," in *Proc. CVPR*, Jun. 2020, pp. 9031–9040.

[29] L. Han, T. Zheng, L. Xu, and L. Fang, "OccuSeg: Occupancy-aware 3D instance segmentation," in *Proc. CVPR*, 2020, pp. 2940–2949.

[30] T. He, C. Shen, and A. van den Hengel, "DyCo3D: Robust instance segmentation of 3D point clouds through dynamic convolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 354–363.

[31] S. Qiu, S. Anwar, and N. Barnes, "Semantic segmentation for real point cloud scenes via bilateral augmentation and adaptive fusion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 1757–1767.

[32] M. Xu, R. Ding, H. Zhao, and X. Qi, "PAConv: Position adaptive convolution with dynamic kernel assembling on point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 1–15.

[33] H. Zhao, L. Jiang, J. Jia, P. Torr, and V. Koltun, "Point transformer," 2020, *arXiv:2012.09164*.

[34] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph CNN for learning on point clouds," *ACM Trans. Graph.*, vol. 38, no. 5, pp. 1–12, 2019.

[35] L. Wang, Y. Huang, Y. Hou, S. Zhang, and J. Shan, "Graph attention convolution for point cloud semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 10296–10305.

[36] K. Fujiwara and T. Hashimoto, "Neural implicit embedding for point cloud analysis," in *Proc. CVPR*, 2020, pp. 11734–11743.

[37] I. Alonso, L. Riazuelo, L. Montesano, and A. C. Murillo, "3D-MiniNet: Learning a 2D representation from point clouds for fast and efficient 3D LiDAR semantic segmentation," *IEEE Robot. Automat. Lett.*, vol. 5, no. 4, pp. 5432–5439, Jul. 2020.

[38] A. Xiao, X. Yang, S. Lu, D. Guan, and J. Huang, "FPS-Net: A convolutional fusion network for large-scale LiDAR point cloud segmentation," *ISPRS J. Photogramm. Remote Sens.*, vol. 176, pp. 237–249, Jun. 2021.

[39] Q. Hu *et al.*, "RandLA-Net: Efficient semantic segmentation of large-scale point clouds," in *Proc. CVPR*, 2020, pp. 11108–11117.

[40] S. Fan, Q. Dong, F. Zhu, Y. Lv, P. Ye, and F.-Y. Wang, "SCF-Net: Learning spatial contextual features for large-scale point cloud segmentation," in *Proc. CVPR*, 2021, pp. 14504–14513.

[41] H. Shuai, X. Xu, and Q. Liu, "Backward attentive fusing network with local aggregation classifier for 3D point cloud semantic segmentation," *IEEE Trans. Image Process.*, vol. 30, pp. 4973–4984, 2021.

[42] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, and M. Bennamoun, "Deep learning for 3D point clouds: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 12, pp. 4338–4364, Dec. 2021.

[43] Y. Yang, C. Feng, Y. Shen, and D. Tian, "FoldingNet: Point cloud auto-encoder via deep grid deformation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 206–215.

[44] L. P. Tchapmi, V. Kosaraju, H. Rezatofighi, I. Reid, and S. Savarese, "TopNet: Structural point cloud decoder," in *Proc. CVPR*, 2019, pp. 383–392.

[45] Z. Huang, Y. Yu, J. Xu, F. Ni, and X. Le, "PF-Net: Point fractal network for 3D point cloud completion," in *Proc. CVPR*, 2020, pp. 7662–7670.

[46] T. Groueix, M. Fisher, V. G. Kim, B. C. Russell, and M. Aubry, "A papier-mache approach to learning 3D surface generation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 216–224.

[47] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, "DeepSDF: Learning continuous signed distance functions for shape representation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 165–174.

[48] J. H. X. Wang and L. Ma, "Exploiting local and global structure for point cloud semantic segmentation with contextual point representations," in *Proc. NeurIPS*, 2019, pp. 4571–4581.

[49] S. Cheng, X. Chen, X. He, Z. Liu, and X. Bai, "PRA-Net: Point relation-aware network for 3D point cloud analysis," *IEEE Trans. Image Process.*, vol. 30, pp. 4436–4448, 2021.

[50] J. Yang *et al.*, "Modeling point clouds with self-attention and Gumbel subset sampling," in *Proc. CVPR*, 2019, pp. 3323–3332.

[51] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[52] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4700–4708.

[53] Y. Liu, B. Fan, G. Meng, J. Lu, S. Xiang, and C. Pan, "DensePoint: Learning densely contextual representation for efficient point cloud processing," in *Proc. ICCV*, 2019, pp. 5239–5248.

[54] X. Wen, T. Li, Z. Han, and Y.-S. Liu, "Point cloud completion by skip-attention network with hierarchical folding," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 1939–1948.

[55] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2117–2125.

[56] S. Qiu, S. Anwar, and N. Barnes, "Dense-resolution network for point cloud classification and segmentation," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2021, pp. 3813–3822.

[57] C. Feichtenhofer, H. Fan, J. Malik, and K. He, "SlowFast networks for video recognition," in *Proc. ICCV*, 2019, pp. 6202–6211.

[58] Y. Qian, J. Hou, S. Kwong, and Y. He, "Deep magnification-flexible upsampling over 3D point clouds," *IEEE Trans. Image Process.*, vol. 30, pp. 8354–8367, 2021.

[59] P. Varma *et al.*, "Multi-resolution weak supervision for sequential data," in *Proc. NeurIPS*, 2019, pp. 192–203.

[60] J. F. Blinn, "Hyperbolic interpolation," *IEEE Comput. Graph. Appl.*, vol. 12, no. 4, pp. 89–94, Jul. 1992.

[61] J. Mao, X. Wang, and H. Li, "Interpolated convolutional networks for 3d point cloud understanding," in *Proc. ICCV*, 2019, pp. 1578–1587.

[62] J. Gong *et al.*, "Boundary-aware geometric encoding for semantic segmentation of point clouds," in *Proc. AAAI*, 2021, pp. 1424–1432.

[63] B. Yang *et al.*, "Learning object bounding boxes for 3D instance segmentation on point clouds," in *Proc. NeurIPS*, 2019, pp. 6737–6746.

[64] W. Li, K. Mueller, and A. Kaufman, "Empty space skipping and occlusion clipping for texture-based volume rendering," in *Proc. IEEE Trans. Ultrason., Ferroelectr., Freq. Control*, Oct. 2003, pp. 317–324.

[65] L. Tchapmi, C. Choy, I. Armeni, J. Gwak, and S. Savarese, "SEGCloud: Semantic segmentation of 3D point clouds," in *Proc. Int. Conf. 3D Vis. (DV)*, Oct. 2017, pp. 537–547.

[66] B.-S. Hua, Q.-H. Pham, D. T. Nguyen, M.-K. Tran, L.-F. Yu, and S.-K. Yeung, "SceneNN: A scene meshes dataset with aNNotations," in *Proc. 4th Int. Conf. 3D Vis. (DV)*, Oct. 2016, pp. 92–101.

[67] W. Wang, R. Yu, Q. Huang, and U. Neumann, "SGPN: Similarity group proposal network for 3D point cloud instance segmentation," in *Proc. CVPR*, 2018, pp. 2569–2578.

[68] X. Wang, S. Liu, X. Shen, C. Shen, and J. Jia, "Associatively segmenting instances and semantics in point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4096–4105.

[69] T. He, Y. Liu, C. Shen, X. Wang, and C. Sun, "Instance-aware embedding for point cloud instance segmentation," in *Proc. ECCV*, 2020, pp. 255–270.

[70] T. He, D. Gong, Z. Tian, and C. Shen, "Learning and memorizing representative prototypes for 3D point cloud semantic and instance segmentation," in *Proc. ECCV*, 2020, pp. 564–580.

[71] J. Liu, M. Yu, B. Ni, and Y. Chen, "Self-prediction for joint instance and semantic segmentation of point clouds," in *Proc. ECCV*, 2020, pp. 187–204.

[72] H. Lei, N. Akhtar, and A. Mian, "Spherical kernel for efficient graph convolution on 3D point clouds," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 10, pp. 3664–3680, Mar. 2020.

[73] Y. Xu, T. Fan, M. Xu, L. Zeng, and Y. Qiao, "SpiderCNN: Deep learning on point sets with parameterized convolutional filters," in *Proc. ECCV*, 2018, pp. 87–102.

[74] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu, "Deeply-supervised nets," in *Proc. 18th Int. Conf. Artif. Intell. Statist.*, 2015, pp. 562–570.

Yu Zheng received the B.S. degree from the Department of Automation, Tsinghua University, China, in 2019, where he is currently pursuing the Ph.D. degree. His research interests include point cloud recognition and object detection in 3D computer vision. He has served as a Reviewer for a number of journals and conferences, including IEEE TRANSACTIONS ON IMAGE PROCESSING, *Pattern Recognition Letters*, CVPR, ICME, and FG.

Xiuwei Xu received the B.Eng. degree from Tsinghua University in 2021, where he is currently pursuing the Ph.D. degree with the Department of Automation. His research interests include data/computation-efficient learning, 3D vision, and robotic systems.

Jie Zhou (Senior Member, IEEE) received the B.S. and M.S. degrees from the Department of Mathematics, Nankai University, Tianjin, China, in 1990 and 1992, respectively, and the Ph.D. degree from the Institute of Pattern Recognition and Artificial Intelligence, Huazhong University of Science and Technology (HUST), Wuhan, China, in 1995. From 1995 to 1997, he worked as a Postdoctoral Fellow with the Department of Automation, Tsinghua University, Beijing, China, where he has been a Full Professor with the Department of Automation since 2003. In recent years, he has authored more than 100 papers in peer-reviewed journals and conferences. Among them, more than 40 papers have been published in top journals and conferences, such as the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, the IEEE TRANSACTIONS ON IMAGE PROCESSING, and CVPR. His research interests include computer vision, pattern recognition, and image processing. He is an IAPR Fellow. He received the National Outstanding Youth Foundation of China Award. He is an Associate Editor of the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE and two other journals.

Jiwen Lu (Senior Member, IEEE) received the B.Eng. degree in mechanical engineering and the M.Eng. degree in electrical engineering from the Xi'an University of Technology, Xi'an, China, in 2003 and 2006, respectively, and the Ph.D. degree in electrical engineering from Nanyang Technological University, Singapore, in 2012. He is currently an Associate Professor with the Department of Automation, Tsinghua University, Beijing, China. His current research interests include computer vision and pattern recognition. He was/is a member of the Image, Video and Multidimensional Signal Processing Technical Committee, the Multimedia Signal Processing Technical Committee, the Information Forensics and Security Technical Committee of the IEEE Signal Processing Society, the Multimedia Systems and Applications Technical Committee, and the Visual Signal Processing and Communications Technical Committee of the IEEE Circuits and Systems Society. He is a fellow of IAPR. He was a recipient of the National Natural Science Funds for Distinguished Young Scholar. He serves as the General Co-Chair for the International Conference on Multimedia and Expo (ICME) 2022 and the Program Co-Chair for the ICME 2020, the International Conference on Automatic Face and Gesture Recognition (FG) 2023, and the International Conference on Visual Communication and Image Processing (VCIP) 2022. He serves as the Co-Editor-in-Chief for *Pattern Recognition Letters* and an Associate Editor for the IEEE TRANSACTIONS ON IMAGE PROCESSING, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, and the IEEE TRANSACTIONS ON BIOMETRICS, BEHAVIOR, AND IDENTITY SCIENCES.